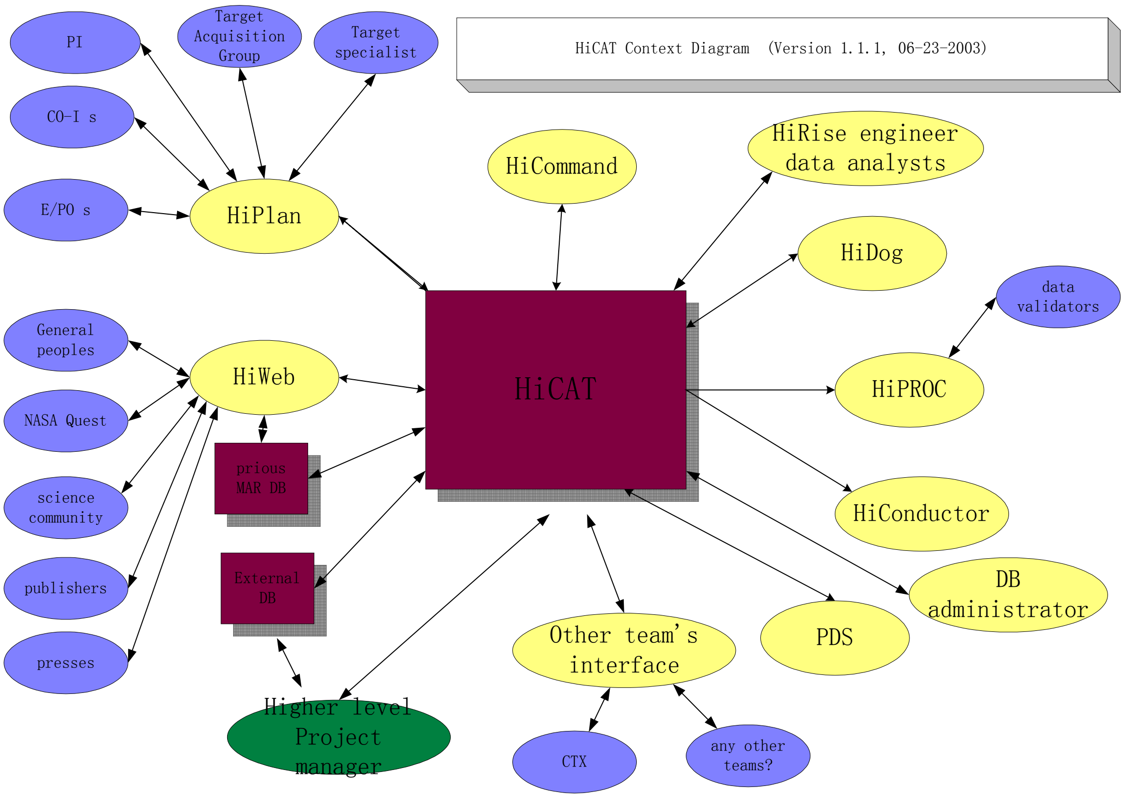
HiCAT Database Design Consideration

Technical Contact: Bradford Castalia Castalia@ pirlmail.lpl.Arizona.EDU
Fei Peng fpeng@pirlmail.lpl.Arizona.EDU

Document History: June 23,2003 -version 1.1.1 1st draft

Based on EOP, HiCAT functional requirements and other relative documents, we provide our draft HiCAT database design consideration as following(Note: the whole life cycle of HiCAT database design is iterative process, we may setup DBMS and create some tables, load some data as prototype to let application programmers to use, at same time, we keep on improving our database design as required):

1. Phase one ---- requirement analysis
    - 1.1 Purpose: try to confirm our understanding of processes, from the highest level to various functional decomposition levels, visually communicate with clients by various DFD(data flow diagrams).
    - 1.2 Information resource(by now):
        - 1.2.1 EOP;
        - 1.2.2 Functional requirements;
        - 1.2.3 PaXX-CFDP accounting and PDS Label File Handling;
        - 1.2.4 Pb.XX—CFDF Science Data File Handling
        - 1.2.5 Any additional information resource?
    - 1.3 Level 1 process analysis –overview of HiCAT
        - 1.3.1 Resource: EOP Figure 5.3.4
    - 1.4 HiCAT context diagram
        - 1.4.1 Purpose: list all potential end user of HiCAT
        - 1.4.2 HiCAT context diagram



External DB = Databases

🟡 = the users(human being users or application users) which interact directly with HiCAT

🔵 = The users who interact with HiCAT indirectly. For example, through HiWEB, HiPlan, HiProc,etc.

🟢

= The higher level users who need to interact with HiCAT and other Database (distributed DBMS) to get necessary information

1.4.3 Questions:

1.4.3.1 Do I miss any potential user types who may interact with HiCAT directly or indirectly? (user type: when retrieve information, update or input information from HiCAT, users who have special needs which are different from other user types)

Please add any additional potential user types.

1.4.3.2 Are the relationship between indirect users and direct users correct?
Please feel free to modify any of the relationship.

1.5 User scenarios description

1.5.1 Purpose: From the point of view of every end user, describe when, what and how he will interact with HiCAT, what information he expect to retrieve, update or input into HiCAT.

1.5.2 Example of user scenarios description

1.5.2.1 Example from other project to show how detail to describe the use case scenarios

Name: activate broadcasting data

Scenarios

(1) Every 60 seconds, timer send broadcast signal to sea buoy controller

(2) Sea buoy controller request wholeData object for current wind, temperature, air, and location data.

(3) If there is SOS signal, Controller passes the signal to transmitter and activate transmitter to broadcast SOS.

If there is request signal, Controller passes the signal to transmitter and activate transmitter to broadcast requested data.

If there is no SOS and request signal, Controller passes current data to transmitter and activate transmitter.

(4) Transmitter broadcasts current data passed by Sea buoy controller.

1.5.2.2 Example 2

User Type: CO-I –in-charge
Scenarios:

(1) A CO-I login into HiPlan. (check login user name and password with HiCAT user profile catalog)
(2) Query the targets requested, can query by requestor, by theme, by date, by status(whether signed priority or not),etc.
(3) Query related Mar observation images stored in previous database
(4) Query related observation plans stored in other DB defined by other teams
(5) Query comments about certain target written by PI, other CO-Is, other HiRISE team members
(6) Query monthly HiRISE meeting records or TAG meeting records about related target.
(7) Input or modify priority of certain target
(8) Input or modify comments about certain target.
(9) Anything else?

1.5.2.3 Need user scenarios description for each end user type(direct users and indirect users)

1.6 DFD (data flow diagram) for each process

1.6.1 Uplink process

1.6.2 Downlink process

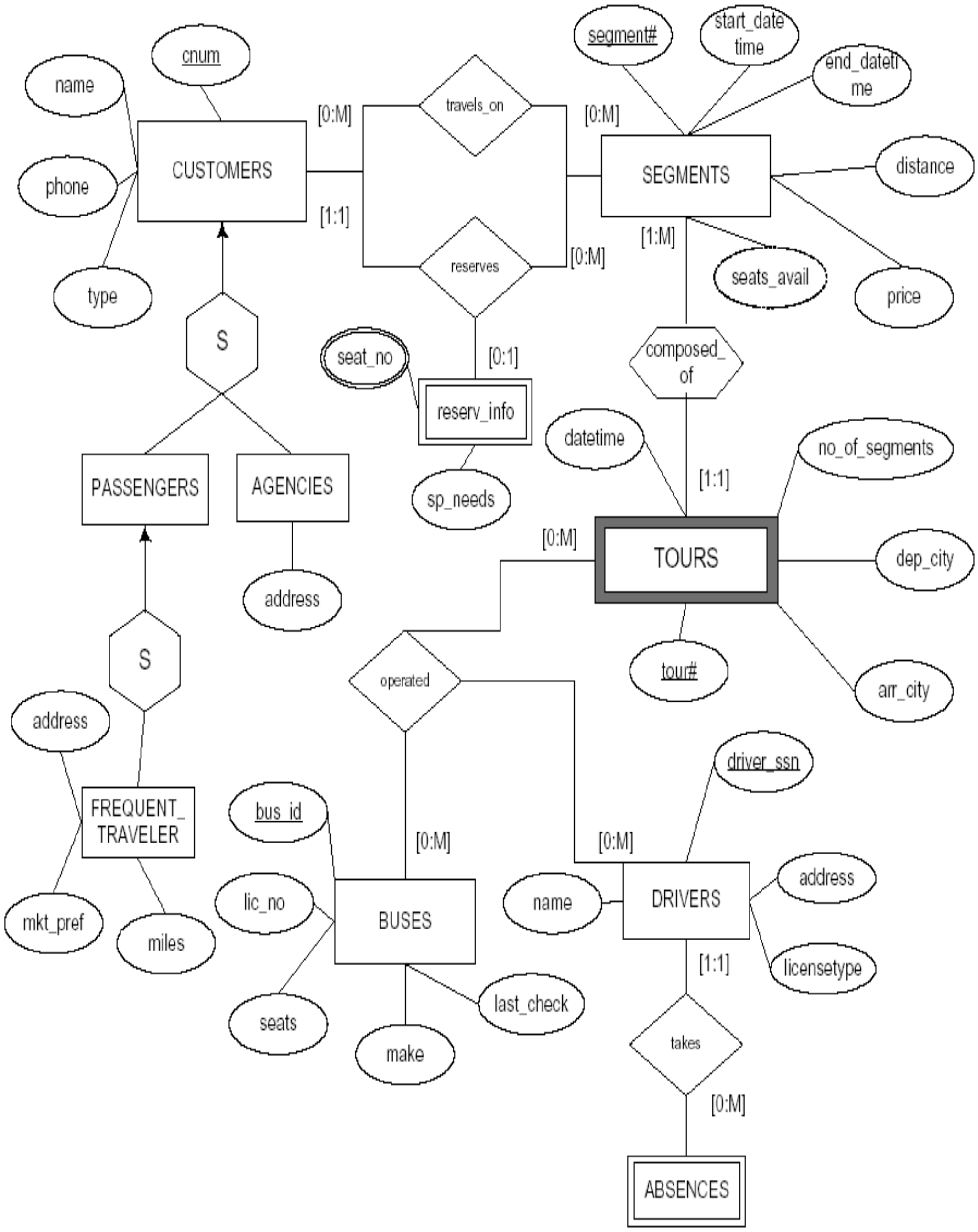
2. Phase 2 ----HiCAT database conceptual design

2.1 Purpose:

Based on above requirement analysis, identify interested entity classes, interested attributes for each entity class, relationship among entity class, identifier for each entity class, mandatory and multi-value of attributes, degree, cardinality and business rules, ensure we get a complete list of entity classes, attributes,etc.

2.2 Deliverable: ERD (entity relational diagram)

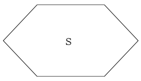
2.3 Example ERD from other project. A complete ERD will look like this:



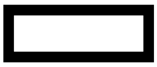
▭ = entity class

⬭ = attribute of entity class

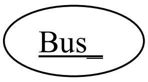
◇ = relationship among entities



= superclass and subclass relationship

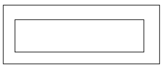


=aggregation relationship

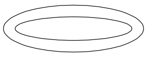


= identifier of an entity class

[1:1]~ [m:m] = cardinality relationship among entities

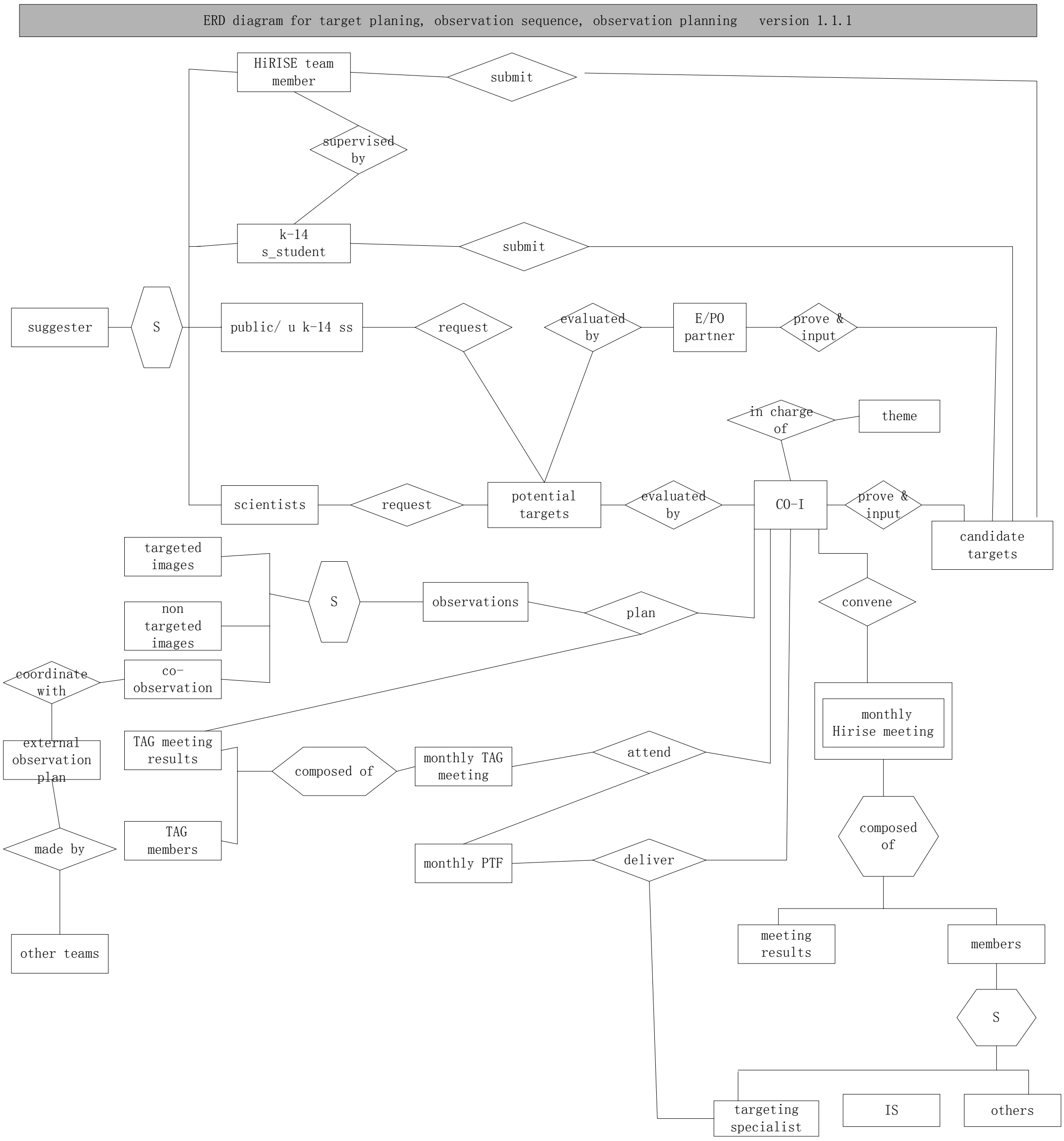


= weak entity



= multi-value entity

2.4 Example ERD for uplink process (rough, incomplete, without attribute, relationship,etc)



2.5 Discussion topics(need communications with end users)

2.5.1: Does the diagram include all potential entity classes you need in HiCAT?

2.5.2 Are the relationship among entity classes correct?

2.5.3 What are the attributes you need to store for each entities in HiCAT?

2.5.4 What are the identifiers for each entity class.

2.5.5 What are the cardinality relationship among entity classes?

2.5.6 Are there any business rules to manipulate each catalog?

2.6 ERD diagram for each processes(such as downlink process, calibration process, image storage process, etc)

3 Phase 3 --- HiCAT database logical design

3.5 Purpose:

Based on above ERD and functional dependency rules, convert ERD to relational schema (database structure, tables). The key point in this process is to ensure a good database design by proceeding normalization analysis to get rid of partial dependency, transitive dependency,get rid of update anomaly, deletion anomaly, insertion anomaly and redundancy problem, ensure the designed tables are in 3NF if needed. Without doing so, it is possible to lead to database performance and integrity problems.

3.6 Deliverable: relational schema (3 NF tables, attributes in tables, identify primary keys and foreign keys)

3.7 Task :

3.3.1 Identify functional dependency

3.3.2 Normalization analysis.

4 Phase 4 ---HiCAT database physical design

4.5 Purpose: create HiCAT database based on above design.

4.6 Deliverable: HiCAT database that operate well.

4.7 Task:

4.7.2 Choose the best DBMS that fit in with HiRise need.

4.7.3 Properly configure DBMS to get the performance, security, integrity as best as possible

4.7.4 Create tables, define data type, data format standards, identify any constraints

4.7.5 Plan and set up HiCAT privilege security system

4.7.6 Plan and implement backup and recovery strategy, replication strategy.

5 Phase 5 --- Testing and Evaluation

5.5 Purpose:

Loading data/sample data into database, test and fine-tunes the database for performance, integrity, concurrence access and security constraints.

5.6 Task:

5.6.2 Fine-tuning specific system and DBMS configuration parameter as needed.

5.6.3 Modify the physical design as needed. (i.e: properly adjust the usage of index to improve performance)

5.6.4 Modify the logical design as needed.

5.6.5 Upgrade or change the DBMS software as needed.

6 Phase 5 ---HiCAT database maintenence and administration

6.5 Purpose:

keep HiCAT operate smoothly, efficiently, with high performance, high security.

6.6 Task:

6.6.2 Moniter HiCAT operation

6.6.3 Moniter and manage user activity
6.6.4 Adjust HiCAT DBMS system environment according to the changing of requirements.
6.6.5 Modify HiCAT database as requirement.
6.6.6 Trouble shooting problems about HiCAT database
6.6.7 Perfective maintenance to enhance the system.